

REMARKS

The final Office Action, mailed June 13, 2006, considered and rejected claims 1-13, 15-18, 22 and 23. Claims 1-4, 6-11, 13 and 15-18 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Ferguson (U.S. Patent No. 6,016,499) in view of Van Huben (U.S. Patent No. 6,484,177). Claims 5 and 12 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Ferguson (U.S. Patent No. 6,016,499) in view of Van Huben (U.S. Patent No. 6,484,177), and further in view of Srinivasan (U.S. Patent No. 6,587,856). Claims 22 and 23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Ferguson (U.S. Patent No. 6,016,499) in view of Glebov (U.S. Patent No. 6,343,265).¹

By this paper, claims 1, 2, 16 and 22 have been amended, claims 5 and 12 cancelled, and no claims added.² Accordingly, following this paper, claims 1-4, 6-11, 13, 15-18, 22 and 23 remain pending, of which claims 1, 16, and 22 are the relevant independent claims at issue.

The pending claims are generally directed to embodiments providing an application with access to data held in a data repository. More particularly, the pending claims are directed to methods and directory interfaces for providing an application access to a repository when the application uses object oriented programming that includes an object class while the repository includes a schema class that is in a different format than the object class. In addition, the pending claims are directed to mapping tools for mapping between the object and schema class.

The method as recited in claim 1, for example, includes receiving from an application, at an interface interposed between the application and the repository, an access command identifying an object class and an object property of the object class in a format specific to the application and that is different than a format utilized by the repository to define a corresponding schema class and schema attribute. The interface also translates the access command to a translated access command, wherein the translated access command identifies the schema class and the schema attribute corresponding to the object class and the object property of the application. Translating the access command includes at least reading a mapping within the object class that identifies the object property of the object class and links the object property of

¹ Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

² Support for the claim amendments can be found throughout Applicant's original application including, by way of example only, the originally filed claims and the disclosure in paragraphs [0053], [0058] and in Figure 5B.

the identified object class to the corresponding schema attribute, wherein the object class is defined by a class definition having therein a definition of the object property and at least one metadata tag associated with the definition of the object property which identifies the schema attribute corresponding to the object property. The translated access command is then transmitted to the repository to obtain access to the repository.

The method recited in claim 16 is directed to a corresponding directory interface and claim 22 is directed to a mapping tool for performing the mapping that is utilized by the directory interface.

While Ferguson and Van Huben are generally directed to embodiments for enabling access to directory service repositories, Ferguson and Van Huben clearly fail to disclose or suggest embodiments for accessing repositories in the claimed manner. For instance, Ferguson and Van Huben fail to disclose or suggest translating access commands to identify a schema class and schema attribute corresponding to the object class and the object property of an application, wherein translating the access command includes reading a mapping within an object class that identifies the object property of the object class and links the object property to the corresponding schema attribute, particularly wherein the object class is defined by a class definition having therein a definition of the object property and at least one metadata tag associated with the definition of the object property., as recited in combination with the other claim elements.

The Examiner appears to acknowledge this deficiency and has turned to the Srinivasan references as purportedly teaching the recited access command translation by using metadata within the application's object class to map between an object property and schema attribute.³

Srinivasan generally relates to method for representing objects in a relational database, and to a system for generating SQL statements for an LDAP search filter. (Col. 4, ll. 30-35). To represent the data, object class tables are used to store object-oriented data in relational tables. (Col. 3, ll. 14-57; Col. 4, ll. 53-61). The rows of the object class tables represent a single object while columns represent object attributes within the object class. (Col. 3, ll. 24-30). To keep track of the data in the system, Srinivasan discloses that metadata is maintained. As noted in Srinivasan, metadata information is "data that describes the structure and parameters of the tables

³ See Office Action, pp. 3, 10. While Van Huben and Ferguson were the only references used in the Office Action for rejecting claims 1 and 16, Srinivasan was combined with Van Huben and Ferguson to reject the now cancelled claim 5, portions of which have been incorporated into claims 1 and 16.

and data maintained in the system." (Col. 6, ll. 27-31). To maintain the metadata, Srinivasan discloses that metadata is maintained within the "same table as its associated data." (Col. 6, ll. 34-36). The various metadata are stored in separate rows within an attribute table. (Col. 6, ll. 36-39).

Figure 5 in Srinivasan illustrates one such attribute table in which metadata (i.e. subschema entries) are defined within the same table that also defines the object class and attributes of the object class. (Col. 7, ll. 4-6). For example, each row having an EID of "2" is a metadata entry. As illustrated, each metadata row describes a type of attribute or object class and lists its properties. (See Fig. 5). Accordingly, Srinivasan discloses that metadata can be used to describe information about object attributes classes.

Srinivasan clearly fails, however, to describe metadata used to read a mapping as recited in the claims. In particular, while Srinivasan discloses that metadata describes information about object attributes or classes, it discloses only metadata that is specific to a single object class. In other words, Srinivasan discloses metadata that describes an object class and attributes of that particular object class, but does not disclose identifying a schema attribute associated with an object property of an object class and which also identifies a schema attribute of a different schema class, as recited in combination with the other claim elements.

The cited references also fail to disclose or suggest an interface or mapping tool for enabling a user to select object classes to be mapped to selectable schema classes and for receiving a user selection of at least one selectable object class and at least one selectable schema class from the graphical user interface, as recited in claim 22. This is particularly true when considering that the selectable object and schema classes are presented in a first graphical user interface and wherein a second graphical interface is provided for user-selection of at least one selectable property of a selected object class and at least one selectable attribute of a selected schema class, in combination with the other recited claim elements.

The Examiner appears to acknowledge that Ferguson fails to teach the recited elements and has turned to Glebov as purportedly teaching the graphical user interface elements that are clearly missing from Ferguson. Applicant respectfully submits, however, that Glebov also fails to disclose or suggest a mapping tool that provides first and second graphical user interfaces, wherein the first enables a user to select object classes to be mapped to selectable schema classes, and wherein the second graphical user interface is provided for user-selection of at least

one selectable property of a selected object class and at least one selectable attribute of a selected schema class, as recited in claim 22 with the other recited elements. This is particularly true when considering the embodiment recited in claim 23, wherein the second interface is only provided after the user first selects the classes from the first interface.

While Glebov generally relates to a mapping tool for object oriented modeling, Glebov fails to disclose a mapping tool as recited in the above claims. In particular, Glebov discloses a mapping system which a developer initially uses a modeling tool 36 to introduce classes, attributes and identify relationships between classes. (Col. 4, ll. 38-49). The mapping tool, as illustrated in Figure 1a, includes a single user interface in which both classes and objects are identified and arrows are used to relate the objects. (Fig. 1a; Col. 4, ll. 41-45).

The modeling tool produces a design model which includes the initial definitions of the objects. (Col. 4, ll. 50-53). The object definitions from the design model are then mapped to metadata that is stored in a common repository. (Col. 4, ll. 51-54). The metadata includes interfaces listing the operations and attributes of the object and mappings to other model constructs. (Col. 4, ll. 60-63). Accordingly, while Glebov discloses that metadata stores mappings to other constructs, and that it stores a listing of attributes, Glebov fails to disclose inserting the metadata within the *definition* of a selected object class as recited in combination with the other claim elements. In fact, it appears that Glebov expressly teaches that the definition is stored in the design model, while the metadata is stored separately in the common repository.

Further, it appears that the mapping tools of Figure 1a, 1b, in which a single interface displays classes and properties/attributes, are the only graphical user interfaces (GUIs) in which a user selects objects, classes or attributes which can then be mapped to other objects and attributes. For example, while additional GUIs are illustrated in Figures 5 and 6, these GUIs do not allow a user to select properties and attributes in response to which metadata is inserted into a definition of a selected object class, as recited in combination with the other claim elements. For example, Glebov notes that with respect to Figure 5, the GUI displays the contents of metadata in which objects *were mapped* by using the GUI of Figure 1. (Col. 7 ln. 64 to Col. 8, ln. 4). Accordingly, the GUI of Figure 5 is a display of previously mapped attributes. Moreover, the GUI in Figure 6 includes a graphical representation of the inheritance relationship *among classes* in which the user can select *to display* inheritance information. (Col. 8, ll. 5-10).

Accordingly, while the GUI in Figure 6 allows user selection, Glebov teaches that the selection is whether to display inheritance information among classes rather than properties and attributes. As only classes are displayed, rather than class properties or attributes, the GUI in Figure 6 fails to act in the claimed manner in which, at the second graphical user interface, user selection is received of a selected object *property* and schema *attribute*, as claimed in combination with the other recited claim elements.

In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 13th day of August, 2006.

Respectfully submitted,



RICK D. NYDEGGER
Registration No. 28,651
JENS C. JENKINS
Registration No. 44,803
COLBY C. NUTTALL
Registration No. 58,146
Attorneys for Applicant
Customer No. 047973